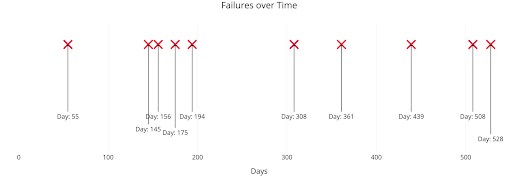
**TOPIC:HDLC queuing MODELS**

**Poisson process**

**A Poisson process, or Poisson point process**, describes a process where certain events occur at a constant rate, but at random and independently of each other.

A Poisson process is a model for a series of discrete events where the average time between events is known, but the exact timing of events is random. The arrival of an event is independent of the event before (waiting time between events is [memoryless](https://en.wikipedia.org/wiki/Memorylessness)).

For example, suppose we own a website that our [content delivery network (CDN)](https://en.wikipedia.org/wiki/Content_delivery_network) tells us goes down on average once per 60 days, but one failure doesn’t affect the probability of the next. All we know is the average time between failures. The failures are a Poisson process that looks like:

Poisson process with an average time between events of 60 days.

We know the average time between events, but the events are randomly spaced in time. We might have back-to-back failures, but we could also go years between failures because the process is stochastic.

A [Poisson process](https://en.wikipedia.org/wiki/Poisson_point_process#Poisson_distribution_of_point_counts) meets the following criteria:

**Poisson Process Criteria**

* Events are independent of each other. The occurrence of one event does not affect the probability another event will occur.
* The average rate (events per time period) is constant.

[Common examples of Poisson processes](https://en.wikipedia.org/wiki/Poisson_distribution#Occurrence) are customers calling a help center, visitors to a website, radioactive decay in atoms, photons arriving at a space telescope and movements in a stock price.

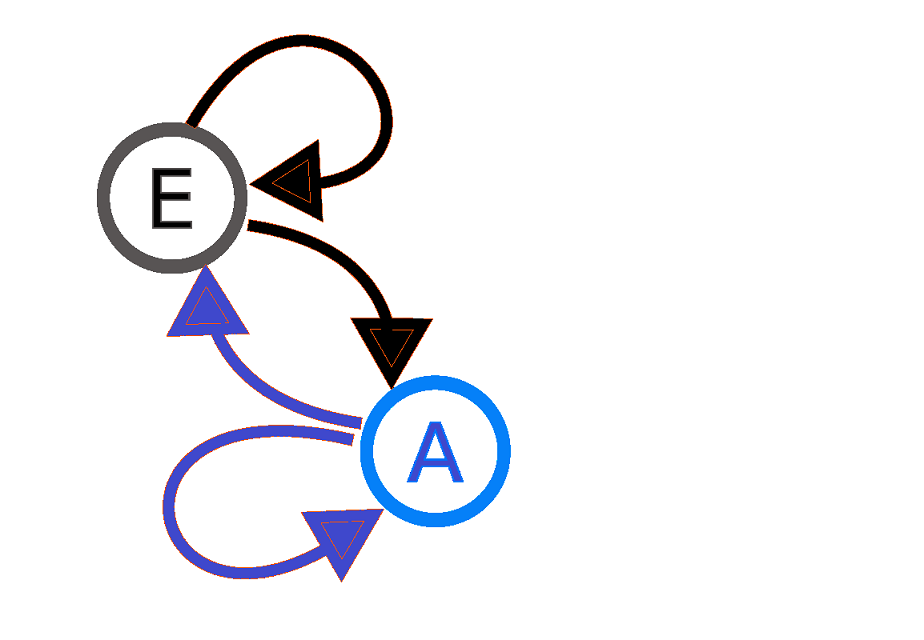
**Topic: Marcov chain**

**Marcov chain:** A **Markov chain** or **Markov process** is a [stochastic model](https://en.wikipedia.org/wiki/Stochastic_process) describing a [sequence](https://en.wikipedia.org/wiki/Sequence) of possible events in which the [probability](https://en.wikipedia.org/wiki/Probability) of each event depends only on the state attained in the previous event. Informally, this may be thought of as, "What happens next depends only on the state of affairs *now*."

**Markov chains.** These are the simplest type of Markov model and are used to represent systems where all states are observable. Markov chains show all possible states, and between states, they show the transition rate, which is the [probability](https://www.techtarget.com/whatis/definition/probability) of moving from one state to another per unit of time. Applications of this type of model include prediction of market crashes, [speech recognition](https://www.techtarget.com/searchcustomerexperience/definition/speech-recognition) and search engine algorithms.

Markov chains, named after **Andrey Markov**, a stochastic model that depicts a sequence of possible events where predictions or probabilities for the next state are based solely on its previous event state, not the states before.

In simple words, the probability that n+1th steps will be x depends only on the nth steps not the complete sequence of steps that came before n. This property is known as Markov Property or Memorylessness. Let us explore our Markov chain with the help of a diagram, are Markov Process



A diagram representing a two-state(here, E and A) Markov process. Here the arrows originated from the current state and point to the future state and the number associated with the arrows indicates the probability of the Markov process changing from one state to another state. For instance, if the Markov process is in state E, then the probability it changes to state A is 0.7, while the probability it remains in the same state is 0.3. Similarly, for any process in state A, the probability to change to Estate is 0.4 and the probability to remain in the same state is 0.6.

**Topic:Queuing theory**

A queueing model is **constructed so that queue lengths and waiting time can be predicted**. Queueing theory is generally considered a branch of operations research because the results are often used when making business decisions about the resources needed to provide a service.

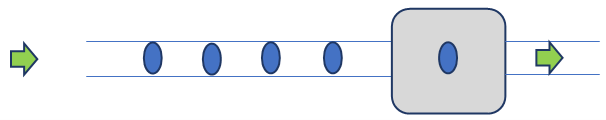
# **M/M/1 Queueing Model**

## ****Basic Concepts****

The **M/M/1 model** is a queueing process in which customers arrive at one server and wait in a queue (if necessary) until the server is available. Customers are serviced in the order in which they arrive (FIFO = first in, first out). The server services at most one customer at a time. There is no limit to the number of customers who can wait in the queue.

In the M/M/1 model, customer arrivals follow an exponential distribution at the rate of λ. Servicing also follows an exponential distribution with a service rate of μ.

The number of customers in the system = the number of customers (if any) waiting in the queue plus one if the server is occupied.

[](https://real-statistics.com/wp-content/uploads/2023/02/queueing-model.png)

**Figure 1 – M/M/1 queueing model**

**Little’s Law** is a theorem that determines the average number of items in a stationary queuing system, based on the average waiting time of an item within a system and the average number of items arriving at the system per unit of time.

The law provides a simple and intuitive approach for the assessment of the efficiency of queuing systems.

### Formula for Little’s Law

Mathematically, Little’s Law is expressed through the following equation:



Where:

L – the average number of items in a queuing system

λ – the average number of items arriving at the system per unit of time

W – the average waiting time an item spends in a queuing system

### Example of Little’s Law

John owns a small coffee shop. He wants to know the average number of customers queuing in his coffee shop, to decide whether he needs to add more space to accommodate more customers. Currently, his queuing area can accommodate no more than eight people.

John measured that, on average, 40 customers arrive at his coffee shop every hour. He also determined that, on average, a customer spends around 6 minutes in his store (or 0.1 hours). Given these inputs, John can find the average number of customers queuing in his coffee shop by applying Little’s Law:

##### **L**  =  40 x 0.1  =  **4 customers**

Little’s Law shows that, on average, there are only four customers queuing in John’s coffee shop. Therefore, he does not need to create more space in the store to accommodate more queuing customers.

**Queueing delay:**   
Let the packet is received by the destination, the packet will not be processed by the destination immediately. It has to wait in a queue in something called a buffer. So the amount of time it waits in queue before being processed is called queueing delay.

In general, we can’t calculate queueing delay because we don’t have any formula for that.

This delay depends upon the following factors:

* If the size of the queue is large, the queuing delay will be huge. If the queue is empty there will be less or no delay.
* If more packets are arriving in a short or no time interval, queuing delay will be large.
* The less the number of servers/links, the greater is the queuing delay.

# M/M/s/K Queueing Model

## ****Basic Concepts****

The **M/M/s/K queueing model** is like the [M/M/1/K model](https://real-statistics.com/probability-functions/queueing-theory/m-m-1-k-queueing-model/), except that there are s servers instead of 1.

It is sufficient to look at the case where s ≤ K (where K is the maximum size of the queue) since if K < s then at most K of the s servers will ever be used. Also note that when K = s, there is no queueing since customers who arrive either get serviced immediately or go away permanently.

**M/G/1 QUEUE**

An **M/G/1 queue** is a queue model where arrivals are **M**arkovian (modulated by a [Poisson process](https://en.wikipedia.org/wiki/Poisson_process)), service times have a **G**eneral [distribution](https://en.wikipedia.org/wiki/Probability_distribution) and there is a single server.